Efficient Solution of Backward Jump-Diffusion PIDEs with Splitting and Matrix Exponentials. *

Andrey Itkin

Numerix LLC,
150 East 42nd Street, 15th Floor, New York, NY 10017, USA

and

Polytechnic Institute of New York University,
6 Metro Tech Center, RH 517E, Brooklyn NY 11201, USA
aitkin@poly.edu

April 12, 2013

Abstract

We propose a new, unified approach to solving jump-diffusion partial integro-differential equations (PIDEs) that often appear in mathematical finance. Our method consists of the following steps. First, a second-order operator splitting on financial processes (diffusion and jumps) is applied to these PIDEs. To solve the diffusion equation, we use standard finite-difference methods, which for multi-dimensional problems could also include splitting on various dimensions. For the jump part, we transform the jump integral into a pseudo-differential operator. Then for various jump models we show how to construct an appropriate first and second order approximation on a grid which supersets the grid that we used for the diffusion part. These approximations make the scheme to be unconditionally stable in time and preserve positivity of the solution which is computed via a matrix exponential. The paper demonstrates that the proposed method is computationally efficient, accurate and simple to implement.

1   Introduction

Partial integro-differential equations (PIDEs) naturally appear in mathematical finance if an underlying stochastic process is assumed to be a combination of diffusion and jumps.

---

*Opinions expressed in this paper are those of the author, and do not necessarily reflect the views of Numerix LLC.

A wide class of Lévy processes fall into this category. In modern popular models such as stochastic volatility or, e.g., hybrid models, jumps could accompany any stochastic factor, thus increasing the overall complexity of the problem. For more details about jump-diffusion processes, see Cont and Tankov (2004), Sato (1999).

Unsurprisingly, most of these PIDEs cannot be solved in closed form. At the same time, a numerical counterpart must be efficient. This is especially important if such a jump-diffusion model is used not only for pricing (given the values of the model parameters), but for their calibration as well. While the solution of the diffusion part (PDE) has been numerously discussed in the literature and various methods were proposed (see, e.g., Andersen and Andreasen (2000), Brennan and Schwartz (1978), Cont and Voltchkova (2003), Duffy (2006), Hout and Foulon (2010), Tavella and Randall (2000)), little can be found for the jump part, which according to the Lévy-Khinchine formula is represented by a non-local integral.

Various methods were suggested to address the construction of an efficient algorithm for solving these type of PIDEs (see, e.g., a discussion of these methods and problems related to their implementation in Carr and Mayo (2007), Itkin and Carr (2012), Strauss (2006)). In particular, they include a discretization of the PIDE that is implicit in the differential terms and explicit in the integral term (Cont and Voltchkova (2003)), Picard iterations for computing the integral equation (d'Halluin et al. (2005, 2004)) and a second-order accurate, unconditionally-stable operator splitting (ADI) method that does not require an iterative solution of an algebraic equation at each time step (Andersen and Andreasen (2000)). Various forms of operator splitting technique were also used for this purpose (Itkin and Carr (2012)). In this paper, we will review operator splitting on financial processes in more detail.

Assuming that an efficient discretization of the PIDE in time was properly chosen, the remaining problem is a fast computation of the jump integral, as it was observed to be relatively expansive. We mention three different approaches to numerical computation of this integral.[1]

The first approach assumes a direct approximation of the integral on an appropriate grid and then applies some standard quadrature method, such as Simpson's rule or Gaussian quadrature. This approach is computationally expensive for two reasons. First, usually the "jump" grid is not the same as the "diffusion" grid. Therefore, after the integral is computed, its values at the jump grid should be re-interpolated to the diffusion grid. Second, the integral is defined on an infinite domain, so either the domain has to be truncated or a non-uniform grid has to be used. Moreover, the complexity becomes greater if an implicit discretization of the integral is used, because it requires the solution of a dense system of linear equations of a large size. Therefore, most often an explicit discretization is utilized, which brings some constraints on the time steps to guarantee stability of the scheme.

However, an exponential change of variables reduces the expense of evaluating the integral at all points. This change converts the integral term into a correlation integral, which can be evaluated at all the grid points simultaneously using a Fast Fourier Transform (FFT). This approach has been suggested by many authors (Andersen and Andreasen (2000), Tavella and

---

[1]For some models it can be computed analytically, so in what follows we do not take these models into account.

Randall (2000), Wilmott (1998)). Still, this could be expensive because a large number of FFT nodes may need to be used for better accuracy. Another issue is that using FFT to compute a product of matrix A and vector requires A to be circulant, while the matrix obtained after discretization of the jump integral is not that type. Therefore, a direct (naive) usage of FFT for this purpose produces undesirable so-called "wrap-around" errors. A common technique to eliminate these errors is to embed A, which is actually a Toeplitz matrix, into a circulant matrix. This, in turn, requires doubling the initial vector of unknowns, which makes the algorithm slower (Itkin and Carr (2012)). Also as noticed in the latter paper for GTSP/CGMY/KoBoL models, the FFT algorithm loses its accuracy when the parameter α of the CGMY model tends to zero.[2]

The second approach to computing the jump integral utilizes an alternative representation of this integral in the form of a pseudo-differential operator, which puts the entire PIDE in the form of a fractional PDE. This problem was considered in Cartea and del Castillo-Negrete (2007) and Itkin and Carr (2012). A recent survey of the existing literature on this subject and techniques for computation of the jump integral using the Grunwald-Letnikov approximation (which is of the first order in space) is given in Andersen and Lipton (2012). As it is known from Abu-Saman and Assaf (2007), Meerschaert and Tadjeran (2004, 2006), Sousa (2008), Tadjeran et al. (2006), a standard Grunwald-Letnikov approximation leads to unconditionally unstable schemes. To improve this, a shifted Grunwald-Letnikov approximation was proposed, which allows construction of an unconditionally stable scheme of the first order in space.[3] However, solving pricing equations to second order in the space variable is almost an industry standard, and therefore this method requires further investigation to address this demand.

The third method[4] exploits a nice idea first proposed in Carr and Mayo (2007). Carr and Mayo found that for some Lévy models, the solution of the integral evolutionary equation[5] is equivalent to the solution of a particular PDE. The problem is then to find a proper space-differential operator (kernel) to construct such a PDE. Carr and Mayo demonstrated the advantage of this approach for the Merton and Kou models, and showed which parabolic equations provide the necessary solution. Later in Itkin and Carr (2012), this idea was further generalized to the class of pseudo-parabolic equations as applied to a class of Lévy processes, known as GTSP/CGMY/KoBoL models. These pseudo-parabolic equations could be formally analytically solved via a matrix exponential. Itkin and Carr then discuss a numerical method for efficiently computing this matrix exponential. When the parameter α of the GTSP/CGMY/KoBoL model is an integer, this method uses a finite-difference scheme similar to those used for solving parabolic PDEs, and the matrix of this FD scheme is banded. Therefore, in this case, the computation of the jump integral:

---

[2]The CGMY model in this limit becomes the VG model (Madan and Seneta (1990)).

[3]A second-order approximation could in principle be constructed as well, but this would result in a massive calculation for the coefficients. Therefore, this approach was not further elaborated on.

[4]We also have to mention one more method proposed by Lipton and Sepp (2009) as applied to the Stern model. Though it is not evident how to generalize this method for other models, it provides a very efficient computational algorithm for the Stern model.

[5]This equation naturally arises at some step of the splitting procedure, if splitting is organized by separating diffusion from jumps.

- Is provided on the same grid as was constructed for the diffusion (parabolic) PDE. Outside of this domain (if ever needed, e.g. for European options), the PIDE grid is further extended to an infinite domain,[6] but no interpolation is required afterwards.

- Has linear $(O(N))$ complexity in the number of the grid nodes N, since the results (e.g., option prices) are given by solving a linear system of equations with a banded matrix. In the case of a real parameter α, Itkin and Carr suggested computing the prices using the above algorithm at three values of an integer $\widetilde{\alpha}$ closest to the given real α, and then interpolating using any interpolation of the second order.

In this paper we use a different flavor of this idea. First, we use an operator-splitting method on the financial processes, thus separating the computation of the diffusion part from the integral part. Then, similar to Itkin and Carr (2012), we represent the jump integral in the form of a pseudo-differential operator. Next we formally solve the obtained evolutionary partial pseudo-differential equations via a matrix exponential. We then show that the matrix exponential can be efficiently computed for many popular Lévy models, and that the efficiency of this method is not worse than that of the FFT. The proposed method is almost universal, i.e., allows computation of PIDEs for various jump-diffusion models in a unified form. We also show that this method is relatively simple to implement.

The rest of the paper is organized as follows. In section 2 we briefly discuss a general form of a backward PIDE for the class of Lévy models. In Section 3, we introduce a splitting technique for nonlinear operators. In section 4, we present our general approach to the solution of the PIDE using a splitting and matrix exponential approach. An explicit construction of various FD schemes of the first and second order is presented in the next section. There we consider the following jump models: Merton, Kou and GTSP (also known as CGMY or KoBoL). The results presented in the last two sections are new, and to the best of our knowledge have not been discussed in the literature. Our technique utilizes some results from matrix analysis related to definitions of M-matrices, Metzler matrices and eventually exponentially positive matrices. In section 6, we present numerical examples that demonstrate the efficiency and accuracy of the proposed method. The final section concludes.

2 Lévy Models and Backward PIDE

To avoid uncertainty, let us consider the problem of pricing equity options written on a single stock. As we will see, this specification does not cause us to lose any generality, but it makes the description more practical. We assume an underlying asset (stock) price S_t be driven by an exponential of a Lévy process

$$S_t = S_0 \exp(L_t), \quad 0 \le t \le T, \tag{1}$$

where t is time, T is option expiration, $S_0 = S_t \mid_{t=0}$, L_t is the Lévy process $L = (L_t)_{0 \le t \le T}$ with a nonzero Brownian (diffusion) part. Under the pricing measure, L_t is given by

$$L_t = \gamma t + \sigma W_t + Y_t, \qquad \gamma, \sigma \in \mathbb{R}, \quad \sigma > 0, \tag{2}$$

---

[6]In other words the PIDE grid is a superset of the corresponding PDE grid.

with Lévy triplet (γ, σ, ν), where W_t is a standard Brownian motion on $0 \le t \le T$ and Y_t is a pure jump process.

We consider this process under the pricing measure, and therefore $e^{-(r-q)t}S_t$ is a martingale, where r is the interest rate and q is a continuous dividend. This allows us to express γ as (Eberlein (2009))

$$\gamma = r - q - \frac{\sigma^2}{2} - \int_{\mathbb{R}} \left(e^x - 1 - x\mathbf{1}_{|x|<1} \right) \nu(dx),$$

where $\nu(dx)$ is a Lévy measure which satisfies

$$\int_{|x|>1} e^x \nu(dx) < \infty.$$

We leave $\nu(dx)$ unspecified at this time, because we are open to consider all types of jumps including those with finite and infinite variation, and finite and infinite activity. [7]

To price options written on the underlying process S_t, we want to derive a PIDE that describes time evolution of the European option prices $C(x,t)$, $x \equiv \log(S_t/S_0)$. Using a standard martingale approach, or by creating a self-financing portfolio, one can derive the corresponding PIDE (Cont and Tankov (2004))

$$rC(x,t) = \frac{\partial C(x,t)}{\partial t} + \left(r - \frac{1}{2}\sigma^2 \right) \frac{\partial C(S,t)}{\partial x} + \frac{1}{2}\sigma^2 \frac{\partial^2 C(S,t)}{\partial x^2}$$
$$+ \int_{\mathbb{R}} \left[C(x+y,t) - C(x,t) - (e^y - 1)\frac{\partial C(x,t)}{\partial x} \right] \nu(dy) \quad (3)$$

for all $(x,t) \in \mathbb{R} \times (0,T)$, subject to the terminal condition

$$C(x,T) = h(x), \tag{4}$$

where $h(x)$ is the option payoff, and some boundary conditions which depend on the type of the option. The solutions of this PIDE usually belong to the class of viscosity solutions (Cont and Tankov (2004)).

We now rewrite the integral term using the following idea. It is well known from quantum mechanics that a translation (shift) operator in L_2 space could be represented as

$$\mathcal{T}_a = \exp\left(a\frac{\partial}{\partial x} \right),$$

so

$$\mathcal{T}_a f(x) = f(x+a).$$

---

[7]We recall that a standard Brownian motion already has paths of infinite variation. Therefore, the Lévy process in Eq.(2) has infinite variation since it contains a continuous martingale component. However, here we refer to the infinite variation that comes from the jumps.

Therefore, the integral in Eq. (3) can be formally rewritten as

$$\int_{\mathbb{R}} \left[C(x+y,t) - C(x,t) - (e^y - 1)\frac{\partial C(x,t)}{\partial x} \right] \nu(dy) = \mathcal{J} C(x,t), \tag{5}$$

$$\mathcal{J} \equiv \int_{\mathbb{R}} \left[\exp\left(y\frac{\partial}{\partial x} \right) - 1 - (e^y - 1)\frac{\partial}{\partial x} \right] \nu(dy).$$

In the definition of operator $\mathcal{J}$ (which is actually an infinitesimal generator of the jump process), the integral can be formally computed under some mild assumptions about existence and convergence if one treats the term $\partial/\partial x$ as a constant. Therefore, operator $\mathcal{J}$ can be considered as some generalized function of the differential operator ∂_x. We can also treat $\mathcal{J}$ as a pseudo-differential operator.

With allowance for this representation, the whole PIDE in the Eq.(3) can be rewritten in operator form as

$$\partial_\tau C(x,\tau) = [\mathcal{D} + \mathcal{J}]C(x,\tau), \tag{6}$$

where $\tau = T - t$ and $\mathcal{D}$ represents a differential (parabolic) operator

$$\mathcal{D} \equiv -r + \left(r - \frac{1}{2}\sigma^2 \right)\frac{\partial}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2}{\partial x^2}, \tag{7}$$

where the operator $\mathcal{D}$ is an infinitesimal generator of diffusion.

Notice that for jumps with finite variation and finite activity, the last two terms in the definition of the jump integral $\mathcal{J}$ in Eq.(3) could be integrated out and added to the definition of $\mathcal{D}$. In the case of jumps with finite variation and infinite activity, the last term could be integrated out. However, here we will leave these terms under the integral for two reasons: i) this transformation (moving some terms under the integral to the diffusion operator) does not affect our method of computation of the integral, and ii) adding these terms to the operator $\mathcal{D}$ negatively influences the stability of the finite-difference scheme used to solve the parabolic equation $\mathcal{D}C(x,t) = 0$. This equation naturally appears as a part of our splitting method, which is discussed in the next section.

3   Operator Splitting Technique

To solve Eq. (6) we use splitting. This technique is also known as the method of fractional steps (see Dyakonov (1964), Samarski (1964), Yanenko (1971)) and sometimes is cited in financial literature as Russian splitting or locally one-dimensionally schemes (LOD) (Duffy (2006)).

The method of fractional steps reduces the solution of the original k-dimensional unsteady problem to the solution of k one-dimensional equations per time step. For example, consider a two-dimensional diffusion equation with a solution obtained by using some finite-difference method. At every time step, a standard discretization on space variables is applied, such that the FD grid contains N_1 nodes in the first dimension and N_2 nodes in the second dimension. Then the problem is solving a system of $N_1 \times N_2$ linear equations, and the

matrix of this system is block-diagonal. In contrast, utilization of splitting results in, e.g., N_1 systems of N_2 linear equations, where the matrix of each system is banded (tridiagonal). The latter approach is easy to implement and, more importantly, provides significantly better performance.

The previous procedure uses operator splitting in different dimensions. Marchuk (1975) and then Strang (1968) extended this idea for complex physical processes (for instance, diffusion in the chemically reacting gas, or the advection-diffusion problem). In addition to (or instead of) splitting on spatial coordinates, they also proposed splitting the equation into physical processes that differ in nature, for instance, convection and diffusion. This idea becomes especially efficient if the characteristic times of evolution (relaxation time) of such processes are significantly different.

For a general approach to splitting techniques using Lie algebras, we refer the reader to Lanser and Verwer (1999). Decomposing the total (compound) operator $\mathcal{L}$ for problems of interest seems natural if, say, $\mathcal{L}$ can be represented as a sum of k noncommuting linear operators $\sum_{i=1}^{k} \mathcal{L}_i$. In this case the operator equation $\mathcal{L}f(t) = \sum_{i=1}^{k} \mathcal{L}_i f(t) = 0$, with $f(t)$ being the unknown dependent variable, can be formally integrated via an operator exponential, i.e.,

$$f(t) = e^{\mathcal{L}t} f(0) = e^{\sum_{i=1}^{k} \mathcal{L}_i t} f(0).$$

Due to the noncommuting property, the latter expression can be factorized into a product of operators

$$f(t) = e^{\mathcal{L}_k} ... e^{\mathcal{L}_1} f(0).$$

This equation can then be solved in N steps sequentially by the following procedure:

$$f^{(1)} = e^{\mathcal{L}_1} f(0),$$
$$f^{(2)} = e^{\mathcal{L}_2} f^{(1)},$$
$$\vdots$$
$$f^{(k)} = e^{\mathcal{L}_k} f^{(k-1)},$$
$$f(t) = f^{(k)}.$$

This algorithm is exact (no bias) if all the operators commute. If, however, they do not commute, the above algorithm provides only a first-order approximation in time (i.e., $O(t)$) to the exact solution.

To get the second-order splitting for noncommuting operators, Strang proposed a new scheme, which in the simplest case ($k = 2$) is (Strang (1968))

$$f(t) = e^{Lt} f(0) = e^{(L_1 + L_2)t} f(0) \approx e^{L_1 \frac{t}{2}} e^{L_2 t} e^{L_1 \frac{t}{2}} f(0) + O(t^2). \tag{8}$$

For parabolic equations with constant coefficients, this composite algorithm is second-order accurate in t provided the numerical procedure that solves a corresponding equation at each splitting step is at least second-order accurate.

For nonlinear operators, the situation is more delicate. As shown in Koch and Thalhammer (2011), the theoretical analysis of the nonlinear initial value problem

$$u'(t) = F(u(t)), \qquad 0 \le t \le T$$

for a Banach-space-valued function $u : [0, T] \to X$ given an initial condition $u(0)$ could be done using calculus of Lie derivatives. A formal linear representation of the exact solution is

$$u(t) = \mathcal{E}_F(t, u(0)) = e^{tD_F} u(0), \qquad 0 \le t \le T,$$

where the evolution operator and Lie derivatives are given by

$$e^{tD_F} v = \mathcal{E}_F(t, v), \quad e^{tD_F} G v = G(\mathcal{E}_F(t, v)), \quad 0 \le t \le T,$$
$$D_F v = F(v), \quad D_F G v = G'(v) F(v)$$

for an unbounded nonlinear operator $G : D(G) \subset X \to X$. Using this formalism, Koch and Thalhammer (2011) showed that Strang's second-order splitting method remains unchanged in the case of nonlinear operators.

Using this result for Eq.(6) gives rise to the following numerical scheme:

$$C^{(1)}(x, \tau) = e^{\frac{\Delta \tau}{2} \mathcal{D}} C(x, \tau), \tag{9}$$
$$C^{(2)}(x, \tau) = e^{\Delta \tau \mathcal{J}} C^{(1)}(x, \tau),$$
$$C(x, \tau + \Delta \tau) = e^{\frac{\Delta \tau}{2} \mathcal{D}} C^{(2)}(x, \tau).$$

Thus, instead of an unsteady PIDE, we obtain one PIDE with no drift and diffusion (the second equation in Eq.(9)) and two unsteady PDEs (the first and third ones in Eq.(9)).

In what follows, we consider how to efficiently solve the second equation, while assuming that the solution of the first and the third equations can be obtained using any finite-difference method that is sufficiently efficient. To this end, in various examples given in the next sections we will explicitly mention what particular method was used for this purpose.

In this paper, we do not discuss the uniqueness and existence of the solution for the PIDE; to do so would move us to the definition of a viscosity solution for this class of integro-differential equations. For more details, see Cont and Tankov (2004) and Arisawa (2005).

Lastly, let us mention that $\mathcal{J} = \phi(-i\partial_x)$, where $\phi(u)$ is the characteristic exponent of the jump process. This directly follows from the Lévy-Khinchine theorem.

4 Solution of a Pure Jump Equation

We begin with the following observation. By definition of the jump generator $\mathcal{J}$, under some mild constraints on its existence, $\mathcal{J}$ could be viewed as a function of the operator ∂_x. Therefore, solving the integral (second) equation in Eq.(9) requires a few steps.

First, an appropriate discrete grid $\mathbf{G}(x)$ has to be constructed in the truncated (originally infinite) space domain. This grid could be nonuniform. An important point is that in the

space domain where the parabolic equations of Eq.(9) are defined, this grid should coincide with the finite-difference grid constructed for the solution of these parabolic equations.[8] This is to avoid interpolation of the solution that is obtained on the jump grid (the second step of the splitting algorithm) to the diffusion grid that is constructed to obtain solutions at the first and third splitting steps.

To make this transparent, let the parabolic equation be solved at the space domain $[x_0, x_k]$, $x_0 > -\infty$, $x_k < \infty$ using a nonuniform grid with $k + 1$ nodes $(x_0, x_1, ..., x_k)$ and space steps $h_1 = x_1 - x_0, ..., h_k = x_k - x_{k-1}$. The particular choice of x_0 and x_k is determined by the problem under consideration. We certainly want $|x_0|$ and $|x_k|$ not to be too large. The integration limits of $\mathcal{J}$ in Eq.(5) are, however, plus and minus infinity. Truncation of these limits usually is done to fit memory and performance requirements. On the other hand, we want a fine grid close to the option strike for better accuracy. Therefore, a reasonable way to construct a jump grid is as follows. For $x_0 \leq x \leq x_k$, the jump grid coincides with the grid used for solution of the parabolic PDEs. Outside of this domain, the grid is expanded by adding nonuniform steps; i.e., the entire jump grid is $x_{-K}, x_{1-K}, ...x_{-1}, x_0, x_1, ..., x_k, x_{k+1}, ..., x_{k+M}$. Here $K > 0$, $M > 0$ are some integer numbers that are chosen based on our preferences. Since contribution to $\mathcal{J}$ from very large values of x is negligible, the outer gridpoints $x_{-K}, x_{1-K}, ...x_{-1}$ and $x_{k+1}, ..., x_{k+M}$ can be made highly nonuniform. One possible algorithm could be to have the steps of these grids be a geometric progression. This allows one to cover the truncated infinite interval with a reasonably small number of nodes.

Second, the discretization of ∂_x should be chosen on $\mathbf{G}(x)$. We want this discretization to:

1. Provide the necessary order of approximation of the whole operator $\mathcal{J}$ in space.

2. Provide unconditional stability of the solution of the second equation in Eq.(9).

3. Provide positivity of the solution.

Let Δ_x denote a discrete analog of ∂_x obtained by discretization of ∂_x on the grid $\mathbf{G}(x)$. Accordingly, let us define the matrix $J(\Delta_x)$ to be the discrete analog of the operator $\mathcal{J}$ on the grid $\mathbf{G}(x)$. The following proposition translates the above requirements to the conditions on $J(\Delta_x)$.

Proposition 4.1 *The finite-difference scheme*

$$C(x, \tau + \Delta\tau) = e^{\Delta\tau J(\Delta_x)} C(x, \tau) \tag{10}$$

is unconditionally stable in time τ and preserves positivity of the vector $C(x, \tau)$ if there exists an M-matrix B such that $J(\Delta_x) = -B$.

Proof By definition of an M-matrix (see Berman and Plemmons (1994)), the class of M-matrices contains those matrices whose off-diagonal entries are less than or equal to zero,

---

[8]So the PIDE grid is a superset of the PDE grid.

while all diagonal elements are positive. All eigenvalues of an M-matrix have a positive real part. Therefore, if B is an M-matrix, all eigenvalues of $J(\Delta_x)$ have a negative real part. Therefore, $\|e^{\Delta\tau J(\Delta_x)}\| < 1$ (in the spectral norm), and thus the scheme Eq.(10) is unconditionally stable.

Now since B is an M-matrix, J is a Metzler matrix (Berman and Plemmons (1994)). An exponential function of the Metzler matrix is a positive matrix. Therefore, if $C(x, \tau)$ is positive, the scheme Eq.(10) preserves the positivity of $C(x, \tau + \Delta\tau)$. $\blacksquare$

This proposition gives us a recipe for the construction of the appropriate discretization of the operator $\mathcal{J}$. In the next section, we will give some explicit examples of this approach.

Once the discretization is performed, all we need is to compute a matrix exponential $e^{\Delta\tau J(\Delta_x)}$, and then a product of this exponential with $C(x, \tau)$. The following facts make this method competitive with those briefly described in the introduction. We need to take into account that:

1. The matrix $J(\Delta_x)$ can be precomputed once the finite-difference grid $\mathbf{G}(x)$ has been built.

2. If a constant time step is used for computations, the matrix $\mathcal{A} = e^{\Delta\tau J(\Delta_x)}$ can also be precomputed.

3. If the above two statements are true, the second splitting step results in computing a product of a matrix with time-independent entries and a vector. The complexity of this operation is $O(N^2)$, assuming the matrix $\mathcal{A}$ is $N \times N$, and the vector is $N \times 1$. However, N in this case is relatively small.[9] One can compare this with the FFT algorithm proposed in Andersen and Andreasen (2000) to compute the correlation integral. This translates into computation of two matrix-by-vector products. This algorithm is $2c \times O(N \log_2 N)$, where c is some coefficient. However, N is relatively high in this case. Typical values are $N = 4096$. On top of that, as was discussed in the introduction to avoid "wrap-around" effects this number must be doubled, i.e., $N = 8192$. Also a post-solution interpolation is required.[10] Finally, for some models (CGMY, VG), the computation of the integral in a neighborhood of $x = 0$ requires special treatment (Cont and Voltchkova (2003)). Overall, the total complexity of FFT with allowance for all these remarks becomes almost the same, or even worse than that of the proposed matrix exponential algorithm.

The above consideration is sufficiently general in the sense that it covers any particular jump model where jumps are modeled as an exponential Lévy process. Clearly, for some models computation of the jump integral can be readily simplified. For instance, Merton's model, which we discuss in the next section, allows another approach with a better performance. Below we discuss this approach in more detail as it seems to be general enough and applicable to some other models as well.

---

[9]Typical values are $N = 100 - 200$.

[10]In more advanced approaches, this step could be eliminated; see Parrot (2009).

5   Examples for Some Popular Models

In this section, we review some popular jump models known in the financial literature. Given a model, our goal is to construct a finite-difference scheme, first for Δ_x, and then for $J(\Delta_x)$, that satisfies the conditions of Proposition 4.1.

5.1   Merton Model

Merton (1976) considered jumps that are normally distributed with the Lévy density

$$\nu(dx) = \lambda \frac{1}{\sqrt{2\pi}\sigma_J} \exp\left[-\frac{(x - \mu_J)^2}{2\sigma_J^2}\right] dx, \tag{11}$$

where λ, μ_J and σ_J are parameters of the model. Considering the pure jump part of the Merton model, one can see that it exhibits finite activity, i.e., a finite number of jumps within any finite time interval. Plugging Eq.(11) into the definition of the operator $\mathcal{J}$ in Eq.(5) and fulfilling a formal integration gives

$$\mathcal{J} = \lambda\left(e^{\mu_J d_x + \frac{1}{2}\sigma_J^2 d_x^2} - \kappa d_x - 1\right), \qquad \kappa = e^{\mu_J + \frac{\sigma_J^2}{2}} - 1, \tag{12}$$

where $d_x \equiv \partial/\partial x$, $d_x^2 \equiv \partial^2/\partial x^2$. The corresponding evolutionary pure jump equation to be solved is

$$C^{(2)}(x, \tau) = \mathcal{A}C^{(1)}(x, \tau), \qquad \mathcal{A} = \exp\left[\lambda\Delta\tau\left(e^{\mu_J d_x + \frac{1}{2}\sigma_J^2 d_x^2} - \kappa d_x - 1\right)\right]. \tag{13}$$

A matrix exponential method for this model with the exponential operator[11] as in Eq.(13) has already been considered in Tangman et al. (2011) using a different derivation (from Carr and Mayo (2007)). They also discuss in more detail various modern methods for computing the matrix exponentials.

Recall that the diffusion equations in Eq.(9) have to be solved up to some order of approximation in time τ. Suppose for this purpose we want to use a finite-difference scheme that provides a second-order approximation, $O((\Delta\tau)^2)$. However, Eq.(13) gives an *exact* solution of the corresponding pure jump equation (the second step in Strang's splitting scheme). Since Strang's scheme guarantees only second-order accuracy $(O((\Delta\tau)^2))$ to the exact solution of the full PIDE, the second step could be computed to the same order of accuracy.

Suppose first that we require only an $O(\Delta\tau)$ approximation. To this end we can use the (1,0) Páde approximation of $e^{\Delta\tau\mathcal{J}}$,

$$e^{\Delta\tau\mathcal{J}} \approx 1 + \Delta\tau\mathcal{J} + O(\Delta\tau),$$

which actually is a well-known Euler explicit scheme. Now the product

$$\mathcal{J}C^{(1)}(x, \tau) = -\lambda(\kappa d_x + 1)C^{(1)}(x, \tau) + \lambda e^{\mu_J d_x + \frac{1}{2}\sigma_J^2 d_x^2}C^{(1)}(x, \tau)$$

---

[11]It is actually a double exponential operator.

can be efficiently computed if one observes that vector

$$z(x, \tau) = e^{\mu_J d_x + \frac{1}{2}\sigma_J^2 d_x^2} C^{(1)}(x, \tau)$$

is a solution of

$$\frac{\partial z(x, s)}{\partial s} = \left(\mu_J d_x + \frac{1}{2}\sigma_J^2 d_x^2 \right) z(x, s)$$

for $0 \leq s \leq 1$ and $z(x, 0) = C^{(1)}(x, \tau)$. This PDE is actually a heat equation and can be solved with complexity $O(N)$. In other words, we just rederived the result and method first obtained in Carr and Mayo (2007).

If one needs a second-order approximation in time, one can use a (1,1) Páde approximation of $e^{\Delta \tau \mathcal{J}}$,

$$e^{\Delta \tau \mathcal{J}} \approx \frac{1 + \frac{1}{2}\Delta \tau \mathcal{J}}{1 - \frac{1}{2}\Delta \tau \mathcal{J}} + O((\Delta \tau)^2),$$

so that $C(x, \tau)$ at this step is a solution of

$$\left(1 - \frac{1}{2}\Delta \tau \mathcal{J} \right) C^{(2)}(x, \tau) = \left(1 + \frac{1}{2}\Delta \tau \mathcal{J} \right) C^{(1)}(x, \tau).$$

Since after discretization of $\mathcal{J}$ is performed, the discrete matrix $J(\Delta x)$ in the right-hand side of this equation is dense, a straightforward attempt to solve this matrix equation is not efficient. However, it could be solved by using Picard iteration.[12] To start the iteration, one needs to compute a product $J(\Delta x)C^{(1)}(x, \tau)$. At every iteration i, one computes a product $J(\Delta x)C_{i-1}^{(2)}(x, \tau)$, where $C_i^{(2)}(x, \tau)$ is the ith approximation of $C^{(2)}(x, \tau)$ obtained after i iterative steps. Both these products can be computed by solving a kind of a heat equation (as was mentioned above) with complexity $O(N)$.

5.2   Kou Model

The Kou model, proposed in Kou and Wang (2004), is a double exponential jump model. Its Lévy density is

$$\nu(dx) = \lambda \left(p\theta_1 e^{-\theta_1 x} \mathbf{1}_{x \geq 0} + (1-p)\theta_2 e^{\theta_2 x} \mathbf{1}_{x < 0} \right) dx, \tag{14}$$

where $\theta_1 > 1$, $\theta_2 > 0$, $1 > p > 0$; the first condition was imposed to ensure that the stock price $S(t)$ has finite expectation. Using this density in the definition of the operator $\mathcal{J}$ in Eq.(5) and carrying out the integration (recalling that we treat $\partial/\partial x$ as a constant) gives

$$\mathcal{J} = \lambda \left[-1 + \mu_0 a + p\theta_1(\theta_1 - a)^{-1} + (1-p)\theta_2(a + \theta_2)^{-1} \right], \tag{15}$$

$$a \equiv \partial_x, \qquad \mu_0 = \frac{p}{\theta_1 - 1} + \frac{1-p}{1 + \theta_2}, \quad -\theta_2 < Re(a) < \theta_1.$$

---

[12]See, for instance, d'Halluin et al. (2005), where conditions were derived on approximation to $\mathcal{J}$ for convergence of these iterations.

The inequality $-\theta_2 < Re(a) < \theta_1$ is an existence condition for the integral defining $\mathcal{J}$ and should be treated as follows: the discretization of the operator $a = \partial_x$ should be such that all eigenvalues of matrix A, a discrete analog of a, obey this condition.

Below we assume that $\lambda > 0$. The construction of the finite-difference scheme in the case $\lambda < 0$ can be done in a similar way.

Let us first consider discretizing a using first-order approximation in h_i, $i = 1, \ldots, N$, where $\{h_i\}$ are the step sizes of the nonuniform space grid. Observe that if $\theta_1 I - A$ is an M-matrix[13], then its inverse is a positive matrix. Therefore, a one-sided forward discretization of a, which we denote as A^F : $\partial C / \partial x = [C(x + h, t) - C(x, t)]/h$, should be chosen for this term. In contrast, for the term $\theta_2 I + A$ to be an M-matrix, a one-sided backward discretization of a, denoted as A^B, on the same grid needs to be applied. As now both $M_1 = p\theta_1(\theta_1 I - A^F)^{-1}$ and $M_2 = (1 - p)\theta_2(\theta_2 I + A^B)^{-1}$ are positive matrices, we need the matrix $M_0 = \mu_0 A - I$ (where A is some discretization of a) to be a Metzler matrix in order to make the entire sum $J(\Delta_x) = \lambda(M_0 + M_1 + M_2)$ a Metzler matrix. As $\mu_0 > 0$, this means that $M_0 = \mu_0 A^F - I$. Thus, by construction, we have proved the following proposition:

Proposition 5.1 *Suppose that discretization of Eq.(15) is done using first-order approximation of a in h_i, $i = 1, \ldots, N$. Also suppose that the following scheme is performed: the operator a in the second and third terms in the right-hand side of Eq.(15) are approximated using one-sided forward differences, and the operator a in the fourth term is approximated using a one-sided backward difference. Then the matrix $J(\Delta_x) = \lambda(M_0 + M_1 + M_2)$, a discrete counterpart of the operator $\mathcal{J}$, is a Metzler matrix.*

According to Proposition 4.1, all that remains to prove is that the diagonal elements of $J(\Delta_x)$ are negative, because then $J(\Delta_x)$ is an M-matrix with a negative sign. In doing so, first observe that both A^F and A^B are triangular with nonzero elements on the diagonal. For simplicity, let us consider a uniform grid in x with step size h. Also let us represent M_1 in the form

$$M_1 = p\theta_1 h(\theta_1 h I - U^F)^{-1},$$

where the matrix U^F is nonzero only on the main diagonal (where all elements are -1) and the first upper diagonal (where all elements are 1). Accordingly, we represent M_2 as

$$M_2 = (1 - p)\theta_2 h(\theta_2 h I + U^B)^{-1},$$

with $U^B = (U^F)^T$. Obviously all eigenvalues of U^F are -1, and those of U^B are 1. Therefore, the diagonal elements of M_1 are $p\theta_1 h/(\theta_1 h + 1)$, and those of M_2 are $(1 - p)\theta_2 h(\theta_2 h + 1)$. Thus, the diagonal elements of $J(\Delta_x)$ are

$$J(\Delta_x)_{i,i} = \lambda\left[-1 - \frac{\mu_0}{h} + p\theta_1\frac{h}{h\theta_1 + 1} + (1 - p)\theta_2\frac{h}{h\theta_2 + 1}\right], \quad i = 1, \ldots, N]. \quad (16)$$

Now the last two terms can be made as small as necessary by choosing a sufficiently small value of h,[14] so that $J(\Delta_x)_{i,i} < 0$ for all $i = 1, \ldots, N$. Thus, by Proposition 4.1, our scheme is

---

[13] I here denotes a unit matrix.

[14] The upper limit for H can be found by equating the right-hand side of Eq.(16) to zero and solving the resulting equation for h.

unconditionally stable, preserves positivity of the solution and is of first-order approximation in h.

The same result can be readily obtained in the case of a nonuniform grid. The discretization of a is constructed in a similar way; instead of a constant grid step size h, variable step sizes h_i should be used.

Again, after an appropriate approximation of ∂_x is chosen, $J(\Delta_x)$ can be precomputed just once.

The following proposition extends this method to construct a second order approximation scheme for the operator $\mathcal{J}$.

Proposition 5.2 *Suppose that the following discretization scheme is performed:*

$$
J(\Delta_x) + \lambda \alpha A_2 = \lambda \left\{ -I + \mu_0 A^C + \alpha A_2 + \frac{p\theta_1}{\theta_1 I - A^F} + \frac{(1-p)\theta_2}{\theta_2 I + A^B} \right.
$$
$$
\left. - \frac{1}{2} h \left[\frac{p\theta_1 (A^F)^2}{(\theta_1 I - A^F)^2} + \frac{(1-p)\theta_2 (A^B)^2}{(\theta_2 I + A^B)^2} \right] \right\}, \quad (17)
$$

where $\alpha > 0$ is a constant, A^C is a centered finite-difference approximation of a of second-order in h on the grid $\mathbf{G}(x)$ and A_2 is a centered finite-difference approximation of $a^2 \equiv \partial_{x,x}$ of second-order in h. Then $J(\Delta_x) + \lambda \alpha A_2$ is the negative of an M-matrix and approximates the operator $\mathcal{J} + \lambda \alpha \partial_{x,x}$ with $O(h^2)$ accuracy.

Proof For simplicity, let us consider a uniform grid with a step size h as generalization for the nonuniform grid is straightforward.

1. *Approximation.* Rewriting Eq.(17) in terms of the operator a and taking a series expansion in h, we find that

$$
\lambda \left\{ -1 + \mu_0 a + \alpha a^2 + \frac{p\theta_1}{\theta_1 - (a + \frac{1}{2} h a^2)} + \frac{(1-p)\theta_2}{\theta_2 + (a - \frac{1}{2} h a^2)} - \frac{1}{2} h \left[\frac{p\theta_1 (a + \frac{1}{2} h a^2)}{(\theta_1 - (a + \frac{1}{2} h a^2))^2} \right. \right.
$$
$$
\left. \left. + \frac{(1-p)\theta_2 (a - \frac{1}{2} h a^2)}{(\theta_2 + (a - \frac{1}{2} h a^2))^2} \right] \right\} = \mathcal{J} + \lambda \alpha \partial_{x,x} + O(h^2),
$$

regardless of what kind of second-order approximation of a^2 in h we chose.

2. *Stability and positivity of the solution.* Observe that under the conditions of Eq.(15), $\theta_1 I - A^F$ is an M-matrix, and therefore its inverse is a positive matrix (see Berman and Plemmons (1994)). Moreover, $(\theta_1 I - A^F)^{-2}$ is also a positive matrix. The same is true for $\theta_2 I + A^B$ and its inverse and inverse squared. Now, according to the previous item, we have freedom in choosing the discrete representation of a^2. Let us rewrite the last three terms in the Eq.(17) in the form

$$
M_3 = h \left\{ \frac{p\theta_1}{\theta_1 h I - U^F} + \frac{(1-p)\theta_2}{\theta_2 h I + U^B} - \frac{1}{2} \left[\frac{p\theta_1 (U^F)^2}{(\theta_1 h I - U^F)^2} + \frac{(1-p)\theta_2 (U^B)^2}{(\theta_2 h I + U^B)^2} \right] \right\},
$$

14

and the first three terms in the form

$$M_4 = -I + \frac{\mu_0}{2h}(U^F + U^B) + \frac{\alpha}{h^2}U^F U^B.$$

As $h \to 0$, the matrix M_3 is positive. That is because $(-U^F)^{-1}$ is upper triangular with all elements on the main and upper diagonals equal to 1, and $(U^B)^{-1}$ is lower triangular with all elements on the main and lower diagonals equal to 1. Therefore, the diagonal elements of M_3 as $h \to 0$ are $d_i = [p\theta_1 + (1-p)\theta_2]/2 > 0$. On the other hand, M_4 is a tridiagonal matrix with negative elements on the main diagonal and positive elements at the first upper diagonal. Since the dominant elements at the first lower diagonal are proportional to $1/h^2$ and positive, it is always possible to choose small enough h such that all elements on this diagonal are positive, and thus the total matrix $J(\Delta_x)$ is the negative of an M-matrix. Then, based on Proposition 4.1, the stability and positivity of the solution follow. ∎

Using Proposition 5.2, it is easy to construct an appropriate scheme for obtaining the solution of the pure jump equation in Eq.(9). As the jump operator in this equation doesn't contain the term $\alpha\lambda\partial_{x,x}$, the trick is to borrow it from the diffusion part. For instance, in the diffusion equation there is a term $\frac{1}{2}\sigma^2\partial_{x,x}$. Therefore, we can split the whole operator $\mathcal{J} + \mathcal{D}$ by leaving half of this term (i.e., $\frac{1}{4}\sigma^2\partial_{x,x}$) in $\mathcal{D}$, while moving the other half to the jump operator. Therefore, the operator in the exponent at the second splitting step becomes $\mathcal{J} + \frac{1}{4}\sigma^2\partial_{x,x}$. A comparison of this expression with the analogous term in Proposition 5.2 immediately gives rise to the explicit representation $\alpha = \sigma^2/(4\lambda)$.

Finding a good choice of the step size h_* is determined by two conditions. The first one requires that all elements of M_4 on the first lower diagonal must be positive. This condition is equivalent to

$$\frac{\alpha}{h_*^2} > \frac{\mu_0}{2h_*}.$$

The second condition is needed to guarantee that all diagonal elements of $J(\Delta_x)$ are negative. This gives

$$-1 - \frac{2\alpha}{h_*^2} + \frac{p\theta_1 + (1-p)\theta_2}{2} < 0.$$

The solution of both inequalities provides the value of h_*.

5.3   CGMY Model

Computation of jump integrals under the CGMY model (also known as the KoBoL model, or more generally as generalized tempered stable processes (GTSPs)) was considered in detail in Itkin and Carr (2012) using a similar approach. GTSPs have probability densities symmetric in a neighborhood of the origin and exponentially decaying in the far tails. After this exponential softening, the small jumps keep their initial stable-like behavior, whereas the large jumps become exponentially tempered. The Lévy measure of GTSPs is given by

$$\mu(y) = \lambda_L \frac{e^{-\nu_L|y|}}{|y|^{1+\alpha_L}}\mathbf{1}_{y<0} + \lambda_R \frac{e^{-\nu_R|y|}}{|y|^{1+\alpha_R}}\mathbf{1}_{y>0}, \tag{18}$$

15

where $\nu_R, \nu_L > 0$, $\lambda_R, \lambda_L > 0$ and $\alpha_R, \alpha_L < 2$. The last condition is necessary to provide

$$\int_{-1}^{1} y^2 \mu(dy) < \infty, \quad \int_{|y|>1} \mu(dy) < \infty. \tag{19}$$

The next proposition follows directly from Proposition 7 of Itkin and Carr (2012).[15]

Proposition 5.3 *The PIDE*

$$\frac{\partial}{\partial \tau} C(x, \tau) = \int_{-\infty}^{\infty} \left[C(x + y, \tau) - C(x, \tau) - \frac{\partial}{\partial x} C(x, \tau)(e^y - 1) \right] \mu(y) dy$$

is equivalent to the PDE

$$\frac{\partial}{\partial \tau} C(x, \tau) = (\mathcal{L}_R + \mathcal{L}_L) C(x, \tau), \tag{20}$$

$$\mathcal{L}_R = \lambda_R \Gamma(-\alpha_R) \left\{ (\nu_R - a)^{\alpha_R} - \nu_R^{\alpha_R} + [\nu_R^{\alpha_R} - (\nu_R - 1)^{\alpha_R}] a \right\},$$

$$\alpha_R < 2, \; Re(\nu_R - a) > 0, \; \nu_R > 1,$$

$$\mathcal{L}_- = \lambda_L \Gamma(-\alpha_L) \left\{ (\nu_L + a)^{\alpha_L} - \nu_L^{\alpha_L} + [\nu_L^{\alpha_L} - (\nu_L + 1)^{\alpha_L}] a \right\},$$

$$\alpha_L < 2, \; Re(\nu_L + a) > 0, \; \nu_L > 0,$$

where Γ is the gamma function and $Re()$ of an operator refers to the spectrum of the discretization of that operator.

In special cases, this equation changes to

$$\mathcal{L}_R = \lambda_R \left\{ \log(\nu_R) - \log(\nu_R - a) + \log\left(\frac{\nu_R - 1}{\nu_R}\right) a \right\} \tag{21}$$

$$\alpha_R = 0, \mathbb{R}(\nu_R - a) > 0, \mathbb{R}(\nu_R) > 1,$$

$$\mathcal{L}_L = \lambda_L \left\{ \log(\nu_L) - \log(\nu_L + a) + \log\left(\frac{\nu_L + 1}{\nu_L}\right) a \right\}$$

$$\alpha_L = 0, \; \mathbb{R}(\nu_L + a) > 0, \; \mathbb{R}(\nu_L) > 0,$$

and

$$\mathcal{L}_R = \lambda_R \left[(\nu_R - a) \log(\nu_R - a) - \nu_R \log(\nu_R) + a \left(\log(\nu_R - 1) - 2\nu_R \coth^{-1}(1 - 2\nu_R) \right) \right]$$

$$\alpha_R = 1, \; Re(\nu_R - a) > 0, \; \nu_R > 1, \tag{22}$$

$$\mathcal{L}_L = \lambda_L \left[(\nu_L + a) \log\left(\frac{\nu_L + a}{\nu_L}\right) - a(1 + \nu_L) \log\left(\frac{\nu_L + 1}{\nu_L}\right) \right]$$

$$\alpha_L = 1, \; Re(\nu_L + a) > 0, \; \nu_L > 0,$$

where the logarithm of the differential operator is defined in the sense of Bakas et al. (1993).

---

[15]In Itkin and Carr's paper, jump integrals were defined on half-infinite positive and negative domains, while here they are defined on the whole infinite domain.

There are a few ways to proceed in this case. First, one can use an extra Strang's splitting; instead of directly solving Eq.(20), solve it in three sweeps. At every step, only one operator, either $\mathcal{L}_R$ or $\mathcal{L}_L$ enters the equation. Thus, the construction of the appropriate discrete operator is simplified. The second approach is based on the observation that eigenvalues of the sum of two M-matrices are also positive. This result follows from Wayl's inequality (see Bellman (1960)). Therefore, if every operator in the right-hand side of Eq.(20) is represented by the negative of an M-matrix, the sum of those operators is also the negative of an M-matrix. However, the discretization of these operators, while on the same grid, could differ, thus adding some flexibility to the construction of the numerical scheme.

As shown in Itkin and Carr (2012), the computation of the matrix exponential could be fully eliminated by using the following approach. First, they show that for $\alpha_I \in \mathbb{Z}$ the solution of the pure jump equation could be reduced to the solution of a system of linear equations where the matrix in the left-hand side of the system is banded. Therefore, the complexity of this solution is $O(N)$. Then to compute the matrix exponential for a real α, first choose three closest values of $\alpha_I \in \mathbb{Z}$. Given the solutions at these α_I, we can interpolate them to give the solution for α. Therefore, the complexity of this solution is that of interpolation.

This approach, however, does not work well if $0 < \alpha < 2$, since we do not have a solution at $\alpha = 2$. To proceed in such a way would then require extrapolation instead of interpolation. It is well known that extrapolation is not a reliable procedure, and so in what follows we apply the general approach of this paper to the GTSP models.

5.3.1 Case 1: $1 < \alpha_R < 2$.

Proposition 5.4 *If $1 < \alpha_R < 2$, then the discrete counterpart L_R of the operator $\mathcal{L}_R$ is the negative of an M-matrix if*

$$L_R = \lambda_R \Gamma(-\alpha_R) \left\{ \left(\nu_R I - A^B \right)^{\alpha_R} - \nu_R^{\alpha_R} + \left[\nu_R^{\alpha_R} - (\nu_R - 1)^{\alpha_R} \right] A^F \right\}.$$

The matrix L_R is an $O(h)$ approximation of the operator $\mathcal{L}_R$.

Proof In this case, $\Gamma(-\alpha_R) > 0$. The last three terms in the curly braces form the negative of an M-matrix. Now, due to the existence conditions in Eq.(20), the step size h must be chosen such that $\nu_R > 1/h$, and therefore $\left(\nu_R - A^B \right)^{\alpha_R}$ is a nonnegative matrix. Therefore, if h also obeys the condition

$$(\nu_R - 1/h)^{\alpha_R} - \nu_R^{\alpha_R} - \left[\nu_R^{\alpha_R} - (\nu_R - 1)^{\alpha_R} \right] 1/h < 0,$$

then L_R is the negative of an M-matrix. But this inequality is always true for $h > 0$. That the approximation is first order follows from the definition of A^F and A^B. Therefore, according to Proposition 4.1, the above scheme is unconditionally stable and preserves the positivity of the solution. ∎

5.3.2 Case 2: $\alpha_R < 0$.

Proposition 5.5 *If $\alpha_R < 0$, then the discrete counterpart L_R of the operator $\mathcal{L}_R$ is the negative of an M-matrix if*

$$L_R = \lambda_R \Gamma(-\alpha_R) \left\{ \left(\nu_R I - A^B\right)^{\alpha_R} - \nu_R^{\alpha_R} + \left[\nu_R^{\alpha_R} - (\nu_R - 1)^{\alpha_R}\right] A^B \right\}.$$

The matrix L_R is an $O(h)$ approximation of the operator $\mathcal{L}_R$.

Proof The proof looks the same as that of Proposition 5.4 if one takes into account that $\nu_R^{\alpha_R} - (\nu_R - 1)^{\alpha_R} < 0$ if $\alpha_R < 0$. ∎

5.3.3 Case 3: $0 < \alpha_R < 1$.

The most difficult case is when $0 < \alpha_R < 1$. This is because $\Gamma(\alpha_R) < 0$ in this range of α_R, so construction of the finite-difference scheme based on Proposition 4.1 experiences some problems. Therefore, we must consider another approach, which is closely related to the concept of an "eventually positive matrix"; see Noutsos and Tsatsomeros (2008). Below we reproduce some definitions from this paper necessary for our further analysis.

Definition An $N \times N$ matrix $A = [a_{ij}]$ is called

- *eventually positive*, denoted by $A \overset{v}{>} 0$, if there exists a positive integer k_0 such that $A^k > 0$ for all $k > k_0$; we denote the smallest such positive integer by $k_0 = k_0(A)$ and refer to $k_0(A)$ as the power index of A;

- *exponentially positive* if for all $t > 0$, $e^{tA} = \sum_{k=0}^{\infty} \frac{t^k A^k}{k!} > 0$;

- *eventually exponentially positive* if there exists $t_0 \in [0, \infty)$ such that $e^{tA} > 0$ for all $t > t_0$. We denote the smallest such nonnegative number by $t_0 = t_0(A)$ and refer to it $t_0(A)$ s the exponential index of A.

We also need the following Lemma from Noutsos and Tsatsomeros (2008):

Lemma 5.6 *Let $A \in \mathbb{R}^{N \times N}$. The following are equivalent:*

1. *A is eventually exponentially positive.*

2. *$A + bI$ is eventually positive for some $b \geq 0$.*

3. *$A^T + bI$ is eventually positive for some $b \geq 0$.*

Based on the above definitions and Lemma 5.6, we can prove the following proposition.

Proposition 5.7 *Suppose $0 < \alpha_R < 1$ and consider the following discrete approximation of the operator $\mathcal{L}_R$:*

$$L_R = \lambda_R \Gamma(-\alpha_R) \left\{ \left(\nu_R I - A^C\right)^{\alpha_R} - \nu_R^{\alpha_R} + \left[\nu_R^{\alpha_R} - (\nu_R - 1)^{\alpha_R}\right] A^B \right\}.$$

Then the matrix L_R (i) is an $O(h)$ approximation of the operator $\mathcal{L}_R$, (ii) has all negative eigenvalues and (iii) is eventually exponentially positive.

Proof We prove each statement separately.

Proof of (i): This follows from the fact that A^C approximates the operator a to second order in h, while ^{B} does the same to first order.

Proof of (ii): As $\Gamma(-\alpha_R) < 0$, the matrix

$$M_1 = \lambda_R \Gamma(-\alpha_R) \left\{ -\nu_R^{\alpha_R} + [\nu_R^{\alpha_R} - (\nu_R - 1)^{\alpha_R}] A^B \right\}$$

is the negative of an M-matrix, with negative elements on the main diagonal and positive elements on the first lower diagonal, while all other elements are zero. The term $M_2 = \lambda_R \Gamma(-\alpha_R) \left(\nu_R I - A^C \right)^{\alpha_R}$ evaluates to a dense matrix with negative elements on the main and first lower diagonal and positive elements on the first upper diagonal. The condition $\nu_R > \lambda_i(A^C)$, $i = 1 \ldots, N$ is always valid since $\mathbb{R}(\lambda_i) = 0$ for $i = 1, \ldots, N$.

Denote the elements on the first lower diagonal of M_1 as $d_{-1}^i(M_1)$, $i = 1, \ldots, N - 1$. Note that $d_{-1}^i(M_2) < 0$ while $d_{-1}^i(M_1) > 0$. Therefore, $d_{-1}^i(M_2 + M_1) < d_{-1}^i(M_1)$. For the elements on the main diagonal (we mark them with index 0) the opposite is true; i.e., $d_0^i(M_2 + M_1) > d_0^i(M_1)$. Also observe that for small enough h, we have $M_1 \propto 1/h$ while $M_2 \propto 1/h_R^\alpha$ and $0 < \alpha_R < 1$. Hence L_R could be made strictly diagonally dominant by a proper choice of h. Moreover, all diagonal elements of L_R are negative. Therefore, according to Gershgorin's circle theorem (Golub and Van Loan (1983)), all eigenvalues of L_R have a negative real part.

Proof of (iii): Observe that L_R has the following properties for sufficiently small values of h:

1. Elements on the main diagonal are negative.

2. Elements on the first lower and upper diagonals are positive.

3. Other elements are small in absolute values as compared with that on the above three diagonals and can have various signs.

To prove the statement of (iii), use Lemma 5.6 to choose b large enough such that elements on the main diagonal of $A + bI$ are all positive. Then $d_0^i(A + bI) > 0$, $d_{-1}^i(A + bI) > 0$ and $d_1^i(A + bI) > 0$ for all $i = 1, \ldots, N$. Also,

$$|(A + bI)_{ij}| \ll |(A + bI)_{ii}|, \forall j > i + 1, \text{or } \forall j < i - 1, \tag{23}$$
$$|(A + bI)_{ij}| \ll |(A + bI)_{i,i-1}|, \forall j > i + 1, \text{or } \forall j < i - 1.$$

Taking the square of $A + bI$ propagates large positive values on the diagonals d_0, d_1, d_{-1} to the diagonals d_2, d_{-2}, so the elements on these diagonals become positive. That is because of the above inequalities in Eq.(23). Therefore, $(A + bI)^{N-1}$ is a positive matrix. Thus, by Lemma 5.6, A is eventually exponentially positive. ∎

Our experiments show that one can choose $\Delta\tau_0 = h$ sufficiently small, as in the proof of Proposition 5.7, to get $e^{\Delta\tau L_R} > 0$ for all $\Delta\tau \geq \Delta\tau_0$. Strictly speaking, the above scheme is not unconditionally stable because of this condition on $\Delta\tau$. However, in contrast to, for example, explicit finite difference schemes, this condition is not restrictive. Indeed, it only sets the lower limit for $\Delta\tau$, while for an explicit FD scheme the convergence condition determines the upper boundary of the time step.

5.3.4 Case 4: $\alpha_R = 0$.

When $\alpha_R = 0$, the CGMY model reduces to the VG model (Madan et al. (1998)). The jump operator $\mathcal{J}_L + \mathcal{J}_R$ is given in Eq.(21). For the following, we rewrite the jump operator as

$$\mathcal{L}_R = \lambda_R \left\{ -\log\left(1 - \frac{a}{\nu_R}\right) + a\log\left(\frac{\nu_R - 1}{\nu_R}\right) \right\}.$$

When ν_R is not too close to 1, it makes sense to move the term $\lambda_R \log\left(\frac{\nu_R - 1}{\nu_R}\right) a$ to the diffusion part $\mathcal{D}_R$, and then approximate it as $J_1 = \lambda_R \log\left(\frac{\nu_R - 1}{\nu_R}\right) A^B$ (to first order approximation in h). This guarantees that if D is an appropriate discrete approximation of $\mathcal{D}$ (e.g., D is the negative of an M-matrix), then $D + J_1$ is also the negative of an M-matrix (because $\log\left(\frac{\nu_R - 1}{\nu_R}\right) < 0$, and thus J_1 is also the negative of an M-matrix).

Once we move the first term in Eq.(21) into the diffusion operator, the second (remaining) term in Eq.(21) could be directly exponentiated to give

$$e^{\Delta\tau\mathcal{J}_{R,2}} = \left(\frac{\nu_R}{\nu_R I - a}\right)^{\Delta\tau\lambda_R}.$$

A discrete approximation of this term to first order in h is

$$Q = e^{\Delta\tau L_{R,2}} = \left(\frac{\nu_R}{\nu_R I - A^F}\right)^{\Delta\tau\lambda_R}. \tag{24}$$

As $(\nu_R I - A^F)$ is an M-matrix, $\frac{\nu_R}{\nu_R I - A^F}$ is a positive upper triangular matrix with all the diagonal elements satisfying $\lambda_{ii} < 1$, $i = 1, \ldots, N$. Accordingly, Q is an upper triangular matrix satisfying the same property for its diagonal elements. Thus, the spectral norm of Q is $\|Q\| < 1$. Therefore, the proposed approximation is unconditionally stable and preserves positivity of the solution.

In the case when ν_R is close to 1, moving J_1 into the diffusion discrete operator D could result in needing a very small h to make the discrete diffusion operator stable. Otherwise, one will face a convection-dominated problem that requires special treatment; see, e.g., Duffy (2006). Therefore, we consider another approach below.

Let us again take the operator $\mathcal{J}_{R,2}$, and recall that $\lambda_R \log\left(\frac{\nu_R}{\nu_R I - A^F}\right)$ is an upper triangular positive matrix. By construction, each nondiagonal element is less than the main diagonal element. This inspires the following representation. Set $B = \frac{\nu_R}{\nu_R I - A^F}$, so[16]

$$\log B = \log\left[b_{1,1}(I + K)\right],$$

where K is an upper triangular matrix with zero elements on the main diagonal and all other elements $k_{i,j} < 1$, $i < j$, $i, j = 1, \ldots, N$. Then

$$\log B = \log(b_{1,1})I + K - \frac{K^2}{2} + \frac{K^3}{3} + \ldots.$$

---

[16]Again for simplicity, we consider only a uniform grid here.

Due to the last condition on the elements of K, this series converges to a positive upper triangular matrix with zeros on the main diagonal. Accordingly, since $0 < b_{1,1} < 1$, the matrix $\log B$ is the negative of an M-matrix. This immediately implies that the discrete operator $J_R = B + J_1$ is the negative of an M-matrix. Therefore, according to Proposition 4.1, the above scheme is unconditionally stable and preserves positivity of the solution.

We have proven the following proposition.

Proposition 5.8 *Suppose $\alpha_R = 0$. Let us discretize the operator $\mathcal{L}_R$ as*

$$L_R = \lambda_L \left[\log\left(\frac{\nu_R - 1}{\nu_R}\right) A^B + \log\left(\frac{\nu_R}{\nu_R I - A^F}\right) \right].$$

Then L_R is an $O(h)$ approximation of the operator $\mathcal{L}_R$, while the whole scheme is unconditionally stable and preserves positivity of the solution.

5.3.5 Approximations of $\mathcal{L}_L$

Approximations to $\mathcal{L}_L$ can be constructed in a way similar to those corresponding to $\mathcal{L}_R$. Below we will present a few propositions that specify our construction. Proofs of these propositions are omitted because they are very similar to that for $\mathcal{L}_R$.

Proposition 5.9 *If $1 < \alpha_L < 2$, then the discrete counterpart L_L of the operator $\mathcal{L}_L$ is the negative of an M-matrix if*

$$L_L = \lambda_L \Gamma(-\alpha_L) \left\{ \left(\nu_L I + A^F\right)^{\alpha_L} - \nu_L^{\alpha_L} + [\nu_L^{\alpha_L} - (\nu_L + 1)^{\alpha_L}] A^B \right\}.$$

L_L is an $O(h)$ approximation of the operator $\mathcal{L}_L$.

Proposition 5.10 *If $\alpha_L < 0$, then the discrete counterpart L_L of the operator $\mathcal{L}_L$ is the negative of an M-matrix if*

$$L_L = \lambda_L \Gamma(-\alpha_L) \left\{ \left(\nu_L I + A^F\right)^{\alpha_L} - \nu_L^{\alpha_L} + [\nu_L^{\alpha_L} - (\nu_L + 1)^{\alpha_L}] A^F \right\}.$$

L_L is an $O(h)$ approximation of the operator $\mathcal{L}_R$.

Proposition 5.11 *Suppose $0 < \alpha_L < 1$ and consider the following discrete approximation of the operator $\mathcal{L}_L$:*

$$L_L = \lambda_L \Gamma(-\alpha_L) \left\{ \left(\nu_L I + A^C\right)^{\alpha_L} - \nu_L^{\alpha_L} + [\nu_L^{\alpha_L} - (\nu_L + 1)^{\alpha_L}] A^F \right\}.$$

Then t matrix L_L (i) is an $O(h)$ approximation of the operator $\mathcal{L}_L$, (ii) has all negative eigenvalues and (iii) is eventually exponentially positive.

Proposition 5.12 *If $\alpha_L = 0$, then the discrete counterpart L_L of the operator $\mathcal{L}_L$ is the negative of an M-matrix if*

$$L_L = \lambda_L \left[\log\left(\frac{\nu_L + 1}{\nu_L}\right) A^F + \log\left(\frac{\nu_L}{\nu_L I + A^B}\right) \right].$$

L_L is an $O(h)$ approximation of the operator $\mathcal{L}_L$.

5.3.6 Second-Order Approximation in h

The second-order approximations of $\mathcal{L}_R$ and $\mathcal{L}_L$ could be constructed in a manner similar to the one described in Proposition 5.2. To give an example, consider the case when $\alpha_R = 0$.

Proposition 5.13 *Suppose that the following discretization scheme is in order:*

$$L_R + \beta A_2 = \lambda_R \left\{ \log\left(\frac{\nu_R - 1}{\nu_R}\right) A^B + \log\left[\frac{\nu_R}{\nu_R I - A^F} - \frac{1}{2}h\frac{\nu_R(A^F)^2}{(\nu_R I - A^F)^2}\right]\right\}, \qquad (25)$$

where $\beta = -\frac{h}{2}\lambda_R \log\left(\frac{\nu_R - 1}{\nu_R}\right)$. Then $L_R + \beta A_2$ is the negative of an M-matrix and approximates $\mathcal{L}_R + \beta\partial_{x,x}$ to $O(h^2)$.

Proof The proof is also similar to that of the Proposition 5.2.

1. *Approximation.* Rewriting Eq.(25) in terms of the operator a and expanding it into a series on h, we find that

$$L_R + \beta A_2 = \lambda_R \left[\log\left(\frac{\nu_R - 1}{\nu_R}\right) a + \log\left(\frac{\nu_R}{\nu_R I - a}\right)\right] + \beta\partial_{x,x} + O(h^2).$$

2. *Positivity and stability of the solution.* Since $(\nu_R I - A^F)$ is an M-matrix, $\frac{\nu_R}{\nu_R I - A^F}$ is a positive matrix. As

$$\frac{1}{2}h\frac{\nu_R(A^F)^2}{(\nu_R I - A^F)^2} = \frac{1}{2}h\frac{\nu_R(U^F)^2}{(\nu_R h I - U^F)^2},$$

the diagonal elements of the matrix

$$H = \frac{\nu_R}{\nu_R I - A^F} - \frac{1}{2}h\frac{\nu_R(A^F)^2}{(\nu_R I - A^F)^2}$$

are

$$\lambda_{i,i}(H) = h\nu_R\frac{\nu_R + 1/2}{(\nu_R + 1)^2}.$$

Thus, a value of h can be found (not too small, but not too large as well)[17] such that H is a positive upper-triangular diagonally dominant matrix. Then $\log H$ is the negative of an M-matrix (see Proposition 5.8).

Thus, $L_R + \beta A_2$ is the negative of an M-matrix as well. ∎

As in Proposition 5.2, we can borrow the term $\beta\partial_{x,x}$ from the diffusion operator when doing Strang's splitting. Again, suppose that in the diffusion equation there is a term $\frac{1}{2}\sigma^2\partial_{x,x}$. Therefore, we can construct splitting of the whole operator $\mathcal{L}_R + \mathcal{D}$ by leaving under $\mathcal{D}$ just a portion η of this term, i.e., $\frac{1}{2}\eta\sigma^2\partial_{x,x}$, and moving the other portion, $\frac{1}{2}(1-\eta)\sigma^2\partial_{x,x}$, to the jump operator. Therefore, the whole operator in the exponent at the second step of splitting becomes $\mathcal{L}_R + \frac{1}{2}(1-\eta)\sigma^2\partial_{x,x}$. Comparison of this expression with the analogous term in Proposition 5.13 gives rise to the explicit representation

$$\eta = 1 - h\frac{\lambda_R}{\sigma^2}\log\left(\frac{\nu_R}{\nu_R - 1}\right).$$

---

[17]This strongly depends on how close ν_R is to 1. Our experiments show that for $\nu_R = 1.1$, a relative step size $h = 0.95$ is fine, while for $\nu_R = 1.01$ this should be changed to $h=1$.

Test	T	K	r	q	C/P	ξ	ρ	κ	θ	λ	μ_J	σ_J
1	1	100	0.05	0.0	C	0.3	-0.5	1.5	0.1	5	0.3	0.1

Table 1: Initial parameters used in test calculations.

6 Numerical Examples

In the first test we used our numerical approach to compute prices of European vanilla options under the Bates model (a jump-diffusion model with Merton's jumps). This solution was compared with the semi-analytical solution obtained by using an inverse Fourier Transform (FFT) since the characteristic function for the Bates model is known in closed form; see, e.g., Crépey (2000).

For the diffusion step we used the method described in detail in Hout and Foulon (2010). A nonuniform space grid was constructed in both x and v dimensions which contained 100 nodes in $x \in [0, S_{max}]$, $S_{max} = 40 \max(S_0, K)$, and 40 nodes in $v \in [0, v_{max}]$, $v_{max} = 5v_0$. Here K is the strike, S_0, v_0 are the initial levels of the stock price and instantaneous variance. For the jump step this grid was extended to $S_{up} = 10^4$. Further increase of S_{up} does not influence the option price much, so this boundary was chosen based on a practical argument. The steps of the jump grid when outside of the diffusion grid (where they both coincide with each other) grew according to geometric progression $h_i = h \times g^i$, where $h = (S_{max} - S_{min})/N$ is an average step size for the diffusion grid, g is the growth factor, which in our experiments was chosen as $g = 1.03$. The total jump grid thus contained 237 nodes, 75 of which were the diffusion grid nodes.

The initial parameters used in the test are given in Table 1.

We computed European option prices under the Bates model in two ways. The first approach utilizes the fact that the characteristic function of the Bates model is known in closed form. Therefore, pricing of European options can be done using any FFT algorithm. Here we used a standard version of the Carr and Madan (1999) method with a constant dumping factor $\alpha = 1.25$ and $N = 8192$ nodes. The second approach (FDE) uses an algorithm described in this paper, i.e., splitting and matrix exponentials, where the diffusion (Heston) equation was solved using the method of fractional steps described in Hout and Foulon (2010).

In Fig. 1 absolute and relative differences in prices obtained in our experiments are presented as a function of moneyness $M = S_0/K$. It is seen that FDE provides a very reasonable accuracy compared to the almost exact solution obtained with the FFT method.

To see how much of the observed numerical error could be attributed to the Heston model itself, e.g., to the FD algorithm for computing a pure diffusion part, we repeated this test with no jumps and presented these results in Fig. 2.

In the second test we considered a model similar to Bates, but with jumps simulated using the VG model. We used the parameters in Table 1. In addition, the VG model parameters were chosen as: $\theta = 0.1, \sigma = 0.4, \nu = 3$, which translates[18] to $\nu_R = 1.5098, \nu_L =$

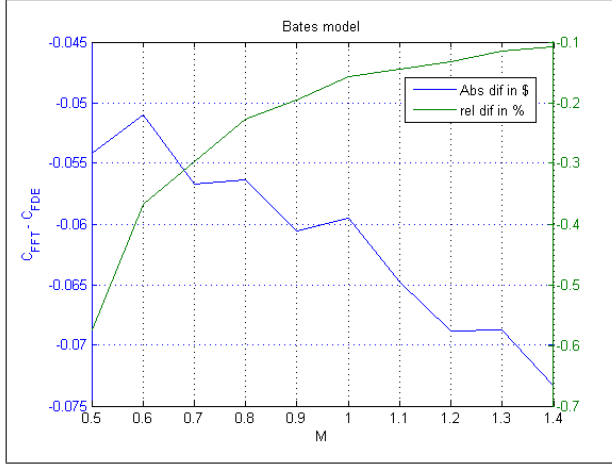[18]For explicit formulae to provide this translation, see Madan et al. (1989).

Figure 1: Absolute and relative differences in call option price as a function of moneyness M for the Bates model computed using an FFT algorithm (FFT) and the algorithm of this paper (FDE).
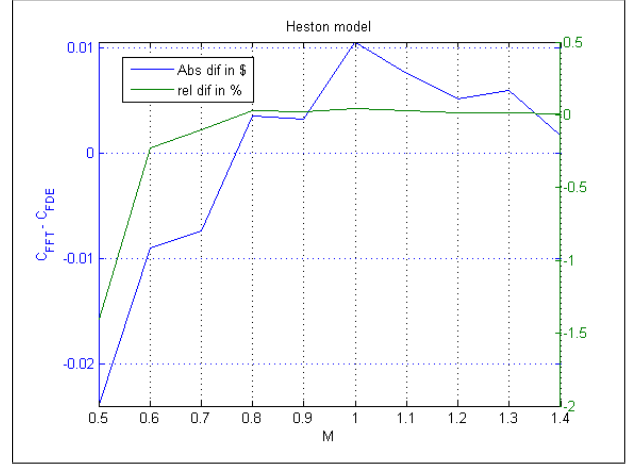
Figure 2: Absolute and relative differences in call option price as a function of moneyness M for the Heston model computed using an FFT algorithm and FDE.

$2.7598, \lambda_R = \lambda_L = 0.33$. The grid was constructed as it was in the previous test. However the upper boundary of the jump grid was moved to 10^5, and $S_{max} = 20 \max(S_0, K)$. Again we computed European option prices in two ways. As the characteristic function of the VG model is known in closed form, the characteristic function of our model is a product of that for the Heston and VG models. We then used an FFT algorithm proposed by Alan Lewis, and as applied to the VG model discussed in detail in Itkin (2005). The second approach uses the FDE algorithm described in this paper.

In Fig. 3, the absolute and relative differences in prices obtained by these two methods are presented as a function of the moneyness $M = S_0/K$. Here FDE behaves worse than in the case of the Bates model, because we used just the first order approximation in h. Still, the relative difference with the exact solution is less than 0.5%, and for $M \approx 0.5$ the difference rises to only 1.7%.

7   Conclusion

In this paper (which is a further extension of our paper Itkin and Carr (2012)) we proposed a new method to solve jump-diffusion PIDEs. This method exploits a number of ideas, namely:

1. First, we transform a linear non-local integro-differential operator (jump operator) into a local nonlinear (fractional) differential operator. Thus, the whole jump-diffusion operator $\mathcal{J} + \mathcal{D}$ is represented as a sum of the linear and non-linear parts.
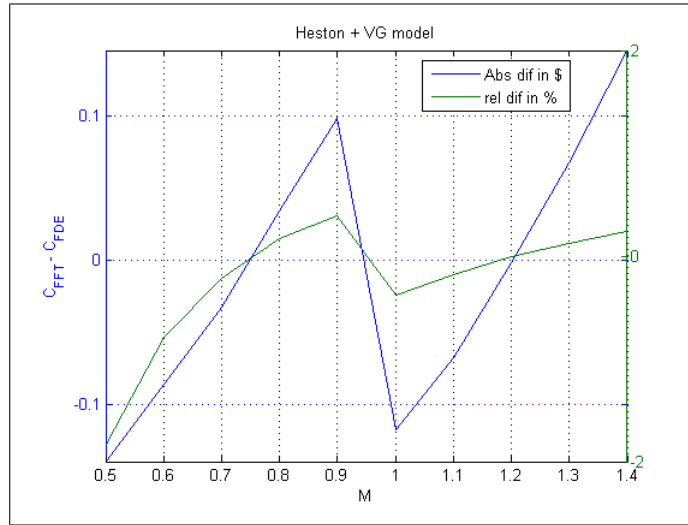
24

Figure 3: Absolute and relative differences in call option price as a function of moneyness M under the Heston+VG model computed using Lewis's FFT algorithm and FDE.

2. Second, operator splitting on financial processes[19] is applied to this operator, namely splitting a space operator into diffusion and jumps parts. For nonlinear operators, this approach was elaborated on based on the definition of Lie derivative (see Koch and Thalhammer (2011)). The described splitting scheme provides a second-order approximation of $\mathcal{J} + \mathcal{D}$ in time.

3. At the third step various finite-difference approximations of the non-linear differential operator $\mathcal{J}$ are proposed for the Merton, Kou and GTSP (a.k.a., CGMY or KoBoL) models. We demonstrated how to construct these approximations to (i) be unconditionally stable, (ii) be of first- and second-order approximation in the space grid step size h and (iii) preserve positivity of the solution. The results are presented as propositions, and the corresponding proofs are given based on modern matrix analysis, including a theory of M-matrices, Metzler matrices and eventually exponentially positive matrices.

Numerical examples in the paper illustrate good efficiency and accuracy of this method. Moreover, the performance of the method is not worse than that of FFT (see the discussion in the introduction).

All these results seem to be new. Also, to the best of our knowledge, all the approaches to solving jump-diffusion PIDE known in the literature were either $O(h)$ or $O(\tau)$, while the proposed methods include finite-difference schemes of the second order approximation, i.e., $O(h^2) + O(\tau^2)$. The method is naturally applicable to both uniform and nonuniform grids,

---

[19]This is similar to splitting on physical processes, e.g., convection and diffusion, which is well-known in computational physics.

and is easy for programming. The method also does not require final reinterpolation (as in the FFT method), because the jump grid includes the PDE grid as a subset.

Also notice that the present approach allows pricing some exotic, e.g., barrier options as well. In addition, it respects not just European (vanilla) payoffs but also digital, American and Bermudan ones.

Acknowledgments

References

Abu-Saman, A. M. and Assaf, A. M. (2007). Stability and convergence of Crank-Nicholson method for fractional advection dispersion equation. *Advances in Applied Mathematical Analysis*, 2(2):117–125.

Andersen, L. and Andreasen, J. (2000). Jump diffusion processes: volatility smile fitting and numerical methods for option pricing. *Review of Derivatives Research*, 4:231–262.

Andersen, L. and Lipton, A. (2012). Asymptotics for exponential Levy processes and their volatility smile: Survey and new results. Available at `http://arxiv.org/abs/1206.6787`.

Arisawa, M. (2005). Viscosity solutions approach to jump processes arising in mathematical finances. In *Proceedings of 10th International conference in mathematical finances.* Also available at `http://www.econ.kyoto-u.ac.jp/daiwa/workshops/2005paper/Arisawa.pdf`.

Bakas, I., Khesin, B., and Kiritsis, E. (1993). The logarithm of the derivative operator and higher spin algebras of w_∞ type. *Communications in Mathematical Physics*, 151(2):233–243.

Bellman, R. (1960). *Introduction to matrix analysis.* McGraw–Hill.

Berman, A. and Plemmons, R. (1994). *Nonnegative matrices in mathmatical sciences.* SIAM.

Brennan, M. and Schwartz, E. (1978). Finite difference methods and jump processes arising in the pricing of contingent claims. *Journal of Financial and Quantitative Analysis*, 13(3):461–474.

Carr, P. and Madan, D. (1999). Option valuation using the fast Fourier transform. *Journal of Computational Finance*, 2:61–73.

Carr, P. and Mayo, A. (2007). On the numerical evaluation of option prices in jump diffusion processes. *The European Journal of Finance*, 13(4):353–372.

Cartea, A. and del Castillo-Negrete, D. (2007). Fractional diffusion models of option prices in markets with jumps. *Physica A*, 374:749–763.

Cont, R. and Tankov, P. (2004). *Financial modelling with jump processes*. Financial Matematics Series, Chapman & Hall /CRCl.

Cont, R. and Voltchkova, E. (2003). A finite difference scheme for option pricing in jump diffusion and exponential Lévy models. Technical Report 513, Rapport Interne CMAP.

Crépey, S. (2000). Computational finance. Évry University, France, available at `http://grozny.maths.univ-evry.fr/pages_perso/crepey/papers/me_crepey.pdf`.

d'Halluin, Y., Forsyth, P. A., and Labahn, G. (2005). A semi-Lagrangian approach for American Asian options under jump diffusion. *SIAM Journal on Scientific Computing*, 27:315–345.

d'Halluin, Y., Forsyth, P. A., and Vetzal, K. R. (2004). A penalty method for American options with jump diffusion processes. *Numerische Mathematik*, 97:321–352.

Duffy, D. (2006). *Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach*. The Wiley Finance Series.

Dyakonov, E. (1964). Difference schemes with a separable operator for general second order parabolic equations with variable coefficient. *Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki*, 4(2):278–291.

Eberlein, E. (2009). Jump-type Lévy processes. In Andersen, T. G., Davis, R. A., Kreiß, J.-P., and Mikosch, T., editors, *Handbook of Financial Time Series*, pages 439–455. Springer Verlag.

Golub, G. and Van Loan, C. (1983). *Matrix Computations*. The Johns Hopkins University Press, Baltimore.

Hout, K. J. I. and Foulon, S. (2010). ADI finite difference schemes for option pricing in the Heston model with correlation. *International journal of numerical analysis and modeling*, 7(2):303–320.

Itkin, A. (2005). Pricing options with VG model using FFT. available at `http://arxiv.org/abs/physics/0503137`.

Itkin, A. and Carr, P. (2012). Using pseudo-parabolic and fractional equations for option pricing in jump diffusion models. *Computational Economics*, 40(1):63–104.

Koch, O. and Thalhammer, M. (2011). Embedded exponential operator splitting methods for the time integration of nonlinear evolution equations. Technical report, Institute for Analysis and Scientific Computing, Vienna University of Technology.

Kou, S. and Wang, H. (2004). Option pricing under a double exponential jump diffusion model. *Management Science*, 50(9):1178–1192.

Lanser, D. and Verwer, J. (1999). Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modelling. *Journal of Computational and Applied Mathematics*, 111(1-2):201–216.

Lipton, A. and Sepp, A. (2009). Credit value adjustment for credit default swaps via the structural default model. *The Journal of Credit Risk*, 5(2):123–146.

Madan, D., Carr, P., and Chang, E. (1998). The variance gamma process and option pricing. *European Finance Review*, 2:79–105.

Madan, D., Milne, F., and Shefrin, H. (1989). The multinomial option pricing model and its Brownian and Poisson limits. *Review of Financial Studies*, 2:251–265.

Madan, D. and Seneta, E. (1990). The variance gamma (V.G.) model for share market returns. *Journal of Business*, 63(4):511–524.

Marchuk, G. (1975). *Methods of Numerical Mathematics*. Springer-Verlag.

Meerschaert, M. M. and Tadjeran, C. (2004). Finite difference approximations for fractional advection-dispersion flow equations. *Journal of Computational and Applied Mathematics*, 172:65–77.

Meerschaert, M. M. and Tadjeran, C. (2006). Finite difference approximations for two-sided space-fractional partial differential equations. *Applied Numerical Mathematics*, 56:80–90.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1):125–144.

Noutsos, D. and Tsatsomeros, M. J. (2008). Reachability and holdability of nonegative states. *SIAM J. Matrix Anal. Appl.*, 30(2):700–712.

Parrot, K. (2009). PDE methods for option pricing under jump diffusion processes. Available at http://www.thalesians.com/archive/public/thalesians/seminars/slides/Thalesians_Parrott_20091118.pdf.

Samarski, A. (1964). Economical difference schemes for parabolic equations with mixed derivatives. *Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki*, 4(4):753–759.

Sato, K.-I. (1999). *Lévy Processes and Infinitely Divisible Distributions*. Cambridge Studies in Advanced Mathematics 68. Cambridge University Press, Cambridge.

Sousa, E. (2008). Finite difference approximations for a fractional advection diffusion problem. Technical Report 08-26, Departamento de Matematica, Universidade de Coimbra.

Strang, G. (1968). On the construction and comparison of difference schemes. *SIAM J. Numerical Analysis*, 5:509–517.

Strauss, A. K. (2006). *Numerical Analysis of Jump-Diffusion Models for Option Pricing.* PhD thesis, Virginia Polytechnic Institute and State University.

Tadjeran, C., Meerschaert, M., and Scheffler, H.-P. (2006). A second-order accurate numerical approximation for the fractional diffusion equation. *Journal of Computational Physics*, 213:205–213.

Tangman, D. Y., Peer, A. A. I., Rambeerich, N., and Bhuruth, M. (2011). Fast simplified approaches to Asian option pricing. *The Journal of Computational Finance*, 14(4):3–36.

Tavella, D. and Randall, C. (2000). *Pricing Financial Instruments. The Finite-Difference Method.* Wiley series in financial engineering. John Wiley & Sons, Inc., New York,.

Wilmott, P. (1998). *Derivatives.* Wiley, New York.

Yanenko, N. (1971). *The method of fractional steps.* Springer-Verlag.